```
//[集合堆疊電腦/The SetStack Computer](1/3)
#define IN "P17IN.txt"
#define OUT "P17OUT.txt"
//***************************************
#include <iostream>
#include <ctime>
using namespace std;
void redir(void);
//***************************************
/* Work Space*/
#include <string>
#include <set>
#include <map>
#include <vector>
#include <stack>
#include <algorithm> //set_union(), set_intersection()
#include <iterator> //inserter()

typedef set<int> Set;
map<Set, int> IdCache; //Set -> ID
vector<Set> SetCache; //ID -> Set

#define ALL(x) x.begin(),x.end()
#define INS(x) inserter(x,x.begin())

int ID(Set x);
//***************************************
```

```
//[集合堆疊電腦/The SetStack Computer](2/3)
int main(void)
{
    redir(); //redirection
//****************************************
/* Work Space*/
    int T, n;
    stack<int> s;//SetID


    scanf("%d", &T);
    while(T--){
        scanf("%d", &n);
        while(n--){
            string op; //block scope
            cin >> op;

            if(op[0] == 'P'){//PUSH
                Set x; //block scope
                s.push(ID(x));
            }else if(op[0] == 'D'){//DUP
                s.push(s.top());
            }else{
                Set x1, x2, x; //block scope
                x1 = SetCache[s.top()];
                s.pop();
                x2 = SetCache[s.top()];
                s.pop();

                if(op[0] == 'U'){//UNION
                    set_union(ALL(x1), ALL(x2), INS(x));
                }else if(op[0] == 'I'){//INTERSECT
                    set_intersection(ALL(x1), ALL(x2), INS(x));
                }else{//ADD
                    x = x2;
                    x.insert(ID(x1));
                }

                s.push(ID(x));
            }
            cout << SetCache[s.top()].size() << endl;
        }
        cout << "***" << endl;
    }
//****************************************
    freopen("CON", "r", stdin); //取消重新導向
    freopen("CON", "w", stdout);

    printf("Time used = %.2f\n", (double)clock()/CLK_TCK); //傳回程式目前為止執行的時間

    system("pause");
    return 0; //the end...
}

void redir(void)
{
    freopen(IN, "r", stdin);
    freopen(OUT, "w", stdout);
}
//****************************************
```

```
//[集合堆疊電腦/The SetS+tack Computer](3/3)
/* Work Space*/
int ID(Set x)
{
    if(IdCache.count(x)){
        return IdCache[x];
    }
    SetCache.push_back(x);
    return IdCache[x] = SetCache.size()-1;
}

//Input(IN) Sample
/*
2
9
PUSH
DUP
ADD
PUSH
ADD
DUP
ADD
DUP
UNION
5
PUSH
PUSH
ADD
PUSH
INTERSECT
*/

//Output(OUT)
/*
0
0
1
0
1
1
2
2
2
***
0
0
1
0
0
***
*/
```